



# TRAVEL TIME ESTIMATION FOR TAXI TRIPS USING GPS SEQUENCE

Abhijeet Tikar<sup>1</sup> | Niranjan Joshi<sup>1</sup> | Manmat Hotalappa<sup>1</sup> | Kajal Gadade<sup>1</sup>

<sup>1</sup> Department of Computer Engineering, NBN Sinhgad School of Engineering, Pune, 411041.

## ABSTRACT

With many companies providing taxi services on a click, request and dispatch of taxi has become easier and accessible to a wide range of people. Predicting travel time of a given taxi trip plays important role in many decision making problems that are important from both the service time and economics perspective of this business. In this problem, we aim at using machine learning techniques to predict trip travel times based on the characteristics of the trip. We use the taxi trip data provided by Kaggle to train and test the data. We discuss our approach based on KNN regression using GPS data available as part of trip trajectories.

## I. INTRODUCTION

With companies like Uber, OLA, request and dispatch of taxi has become easier and accessible to a wide range of people. The problem of automatic taxi assignment to a given trip request is important from both, the service time and economics, perspective of this business. One of the important aspect of this decision is to determine which near-by taxi can be available early and hence predicting trip times becomes important to understand.

The travel time of the trip depends on several factors such as origin, destination, the path/trajectory from origin to destination and more importantly real-time traffic conditions. Traffic conditions in turn are specific to day and/or time of the travel as well as the travel route. Driver and taxi profiles might also affect the travel time. Some drivers might have tendency to drive faster compared to some others. Some taxis might be well-maintained and hence may have good impact of travel time, whereas some other which are poorly maintained might have problems with running at high speed even if the road is empty. Understanding dependency of travel time on these factors is an important study before we actually try to solve the prediction problem here

With the advances in machine learning and recent advances in deep neural networks, it is becoming easier to build prediction models that are robust and yet can perform in realtime. Though training these models is still computationally intensive, using built prediction model in real-time is becoming common to perform prediction tasks. In this problem, we aim at using machine learning techniques to predict trip travel times based on the characteristics of the trip

The competition to solve similar problem was hosted by [5] in affiliation to ECML/PKDD in year 2015-16. The competition was based on Taxi trip travels in Porto in the year 2014-15. The dataset provided as a part of the competition has information of approximately 1.7 billion trips. Several contestants participated and have developed their models to address the problem. We present here the study of already developed models and the scope of improvement to come up with better results. We plan to take on this challenge to further improve the model by addressing some of the aspects that are not covered by existing models.

The competition hosted by [5] consisted of two problems. The first one was to predict the destination of a given taxi trip based on the partial trajectory and other trip information. The second one which we are trying to solve here was to predict the travel time of a given taxi trip based on the starting time, partial trajectory and other trip information such as day of the trip, taxi used etc. Next we describe the dataset that is made available as a part of this competition. Training data containing the following parameters for each taxi trip.

- **TRIP ID:** (String) It contains a unique identifier for each trip;
- **CALL TYPE:** (char) It identifies the way used to demand this service. It may contain one of three possible values:
  - A if this trip was dispatched from the central;
  - B if this trip was demanded directly to a taxi driver on a specific stand;
  - C otherwise (i.e. a trip demanded on a random street).
- **ORIGIN CALL:** (integer) It contains a unique identifier for each phone number which was used to demand, at least, one service. It identifies the trips customer if CALL TYPE=A. Otherwise, it assumes a NULL value;
- **ORIGIN STAND:** (integer): It contains a unique identifier for the taxi

stand. It identifies the starting point of the trip if CALL TYPE=B. Otherwise, it assumes a NULL value;

- **TAXI ID:** (integer): It contains a unique identifier for the taxi driver that performed each trip;
- **TIMESTAMP:** (integer) Unix Timestamp (in seconds). It identifies the trips start;
- **DAYTYPE:** (char) It identifies the daytype of the trips start. It assumes one of three possible values:
  - B if this trip started on a holiday or any other special day (i.e. extending holidays, floating holidays, etc.);
  - C if the trip started on a day before a type-B day;
  - A otherwise (i.e. a normal day, weekday or weekend).
- **MISSING DATA:** (Boolean) It is FALSE when the GPS data stream is complete and TRUE whenever one (or more) locations are missing
- **POLYLINE:** (String): It contains a list of GPS coordinates (i.e. WGS84 format) mapped as a string. The beginning and the end of the string are identified with brackets (i.e. [ and ], respectively). Each pair of coordinates is also identified by the same brackets as [LONGITUDE, LATITUDE]. This list contains one pair of coordinates for each 15 seconds of trip. The last list m corresponds to the trips destination while the first one represents its start; As mentioned earlier the goal is to predict destination and travel time of the trip. For test data destination prediction, output is of the form (TRIP ID, LATITUDE, LONGITUDE) For travel-time, it is of the form (TRIP ID, TRAVEL TIME)

Next, we describe the previous work done with respect to this problem in section II. Section III describes our approach based on knn regression and sampling of trajectory. The results of our approach are discussed in section IV. In the end, we conclude summarizing our work with directions for future work in section V.

## II. LITERATURE SURVEY

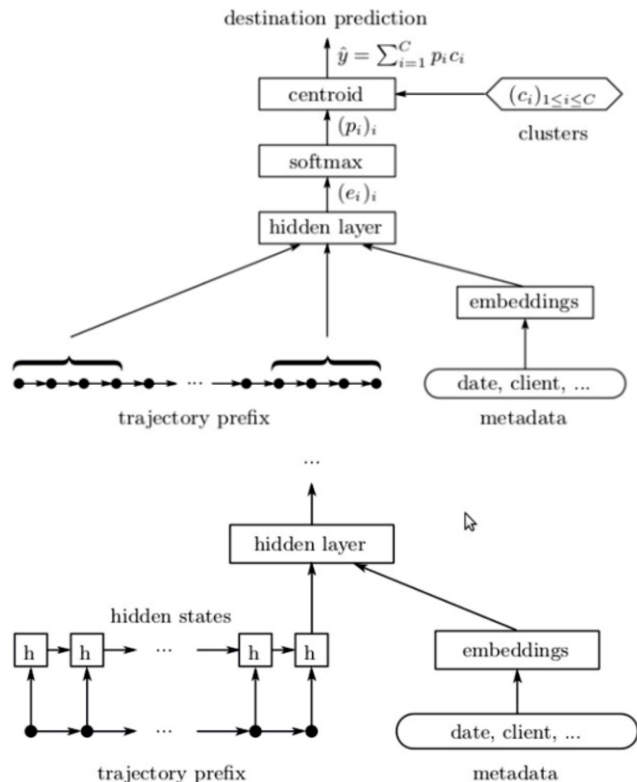
Several teams participated in the competition and reported their model and results achieved using their models. [7] summarizes the competition, dataset, participation and the results. Next, we present here relevant work published through this competition. Since, knowing destination of the trip can also help (in fact, that is a prerequisite) for travel time prediction, we are also including the models used by destination prediction as part of this review.

### A. ANN based approach

The winner team for destination prediction challenge, [1] used a model based on Artificial neural networks with multilayer perceptron architecture (MLP). Figure 1 shows the model. Since, the provided dataset consists of varying length data in POLY LINE field, but MLP requires fixed length input, they used first and last k GPS points in each POLY LINE as input to the model. This gives 2k GPS points or 4k numerical values (longitude and latitude for each GPS point). The approach also used metadata by creating an embedding for each metadata field (such as day-of-the-week, origin, taxi stand etc). The embeddings along with 4k numerical values obtained above form the feature space. Each input (taxi trip) is represented in this feature space. For the destination prediction, the destinations were first grouped into few thousand clusters using mean-shift clustering algorithm. A weighted average of centers of these clusters was used to predict the

final destination. Since, the model didn't train very well using Haversine distance function, a simpler equirectangular distance was used. Stochastic Gradient descent with momentum was used to minimize mean equirectangular distance between predictions and actual destination points. This was mostly an automated approach and didn't require any manual processing. The network consisted of 500 ReLu neurons.

The paper also describes alternative approaches using Recurrent Neural Network (RNN) and Bi-directional RNN (BRNN) as shown in Figure 2 and Figure 3 respectively.



### B. Kernel Regression method

The other team [6] used Kernel Regression (KR) method for solving the problem. This work involved some preprocessing on data especially to figure out missing data in polyline by observing large jumps between two consecutive GPS locations. The typical features used for Kernel Regression method here included full trajectory, last d meter of trajectory, Euclidean and haversine distance between end-points of trajectory, direction of movement (going in-the-city vs going out-of-the-city). This approach also used the contextual features which included day-of-the-week, taxi id, call id, taxi stand etc wherever it is available. To counter the sensitivity of KR prediction performance due to influence of noisy GPS updates, the trips were simplified using RDP algorithm. For travel time prediction, additional features such as average speed, average acceleration and shape complexity (ratio of euclidean distance to haversine distance between end-points) were also considered. To speed up the process of feature extraction, an index structure based on geohash was used. Each GPS was represented using its geohash and then the nearest trips within the maximum distance threshold of 1km were searched using range queries.

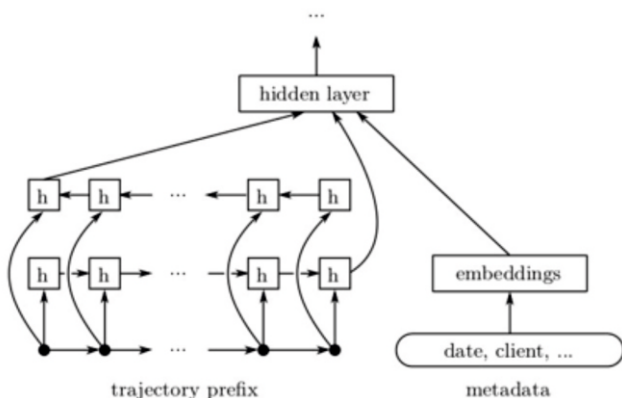


Fig. 3: Bidirectional RNN architecture.

Support Vector Regression (SVR) also yielded similar results. For trip time prediction, ensemble of following regression models was used

- Gradient Boosted Regression Trees (GBRT)
- Random Forest regressor (RF)
- Extremely Randomized Tree regressor (ERT) Stacked generalization approach was used to produce a single predictor. For destination prediction, Mean Haversine Distance (MHD) was used as an error measure of performance, whereas Root Mean Square Logarithmic Error was used for the same in case of travel prediction problem.

### C. Ensemble learning approach

The winner team for travel time prediction [3], used an ensemble learning approach. The framework here consists of hierarchy of expert models as given below

- Expert models for each test trip (e.g. trained on tracks which cross the test trip at the last known position).
- General base model: Based on a data set, where the features were extracted from all the tracks in the training set, and longer tracks were sampled more frequently than shorter ones.
- General expert models for short trips (e.g. only 1, 2 or 3 positions of the initial trajectory are known).

The features used in this approach included haversine and cumulative distance of start and current positions from city center, median velocity, heading of the car in current position etc. i Either a random forest regression or a gradient boosting regression has been used as base classifier for ensemble modeling. Generally, bayesian optimization is used on a hold-out test set to tune weight factors. But, for this work, following heuristics were used –

- Use expert model for all test trips with sufficiently large training set.
- Use Average of all four models for all other test trips. Training sets were generated differently for different models as described below
- For base model, training set contained all trips.
- For expert model of short trips, a separate training set was built using first few GPS readings of all trips for each expert corresponding to trip length.
- For expert models for each test trip, a spatial clustering approach was used and all trips close to the current GPS position of taxi were selected. RMSLE was used as error measure for travel time prediction, whereas MHD was used for destination prediction. The key conclusion from this is that the remaining traveling time of a taxi depends mainly on the current position and heading of the taxi.

### D. Trajectory distribution based approach

[2] uses a trajectory distribution based approach for destination prediction. This approach involves modeling of traffic flow pattern as a mixture of 2d gaussian distributions. Known trajectories are then clustered using hierarchical clustering with ward-linkage criterion based on the Symetrized Segmentpath- Distance. This distance compares trajectories as a whole, regardless of their time indexing or the number of locations that compose them. A new trajectory is then assigned to one of the clusters to predict the final destination. To predict the destination of new trajectory, only beginning of its path is observed for a succession of locations. Contextual information such as hour-of-the-day, day-of-the-week is considered in the prediction model using auxiliary weights which is calculated as the product of any combination of three types of weights given below.

- Empiric Weight which describes the distribution information of trajectory cluster
- Weekday weight which describes distribution information of the trajectory cluster at a given day of the week.
- Hours weight which describes the distribution of information of the trajectory cluster at a given hour of the day.

### E. Tools and APIs for machine learning

There are number of tools which provide access to standard libraries of machine learning algorithms. [4] is one such popular tool which is being widely used for deep neural networks. R provides many libraries for statistical computing and analysis. These tools can be used for developing new models for this problem. Matlab, scilab, octave provide machine learning toolboxes. Weka is one another tool equipped with number of machine learning algorithms for classification and clustering. [1] used Theano, Block and Fuel for their implementation. Since, training phase is computationally expensive and takes longer, most of the times, people have been using GPUs to train such models. With GPUs, the training time reduces considerably from few days to few weeks. [1] have used Nvidia GPUs for training the model

### III. OUR APPROACH

#### A. Features Used

We experimented with following features

1. Prefix (first k points) of trajectory data
2. Suffix (last k points) of trajectory data
3. Sampled (k points on regular interval) trajectory data

For each of these features we experimented with the model by choosing different sizes  $k = 25, 50, 75$  and  $100$  points. For each point in prefix and suffix we have two features one for each co-ordinate  $x$  and  $y$ , thus giving  $2*k$  features points. When using sample trajectories each point represented as a triple  $(t_i, x_i, y_i)$  where  $t_i$  represents time span for  $i$ th location of trajectory.

This gives  $3*k$  feature points per trajectory. For the short trips less than  $k$  points we used the last point to fill the unavailable locations. Each point in the trajectory was normalized with respect to the co-ordinates of Porto city to avoid Closeness between the two GPS points.

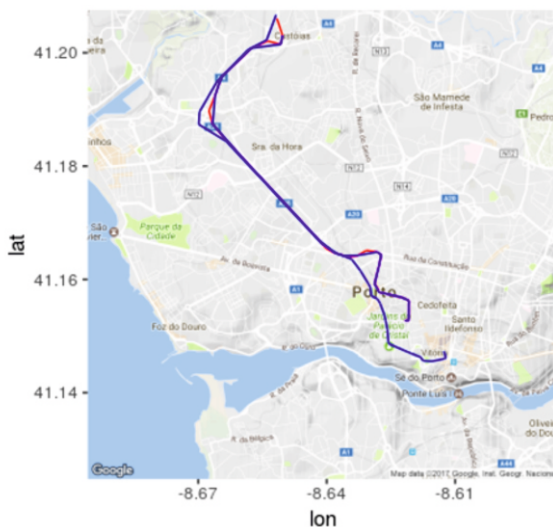


Figure 1: Comparison of Full Trajectory Vs Sampled Trajectory

#### B. Training

We use KNN regression method (knn reg from package caret in R) as our primary training algorithm.

We used 20K trips from the original training dataset for training our model.

We also tried upsampling of training data to ensure that trips of different sizes have enough training instances.

#### C. Location clustering

We also tried clustering GPS points into locations and then mapping each GPS point of trajectory to nearest location which was then used as a feature to the travel time estimation. We used 1 lac GPS points from trajectory data and clustered them using k-median algorithm to obtain 300 clusters representing different locations in the city. Each GPS point in the trajectory was then mapped to centroid of the cluster closest from it. Figure shows the GPS locations in red and cluster formed with their centroid in blue.

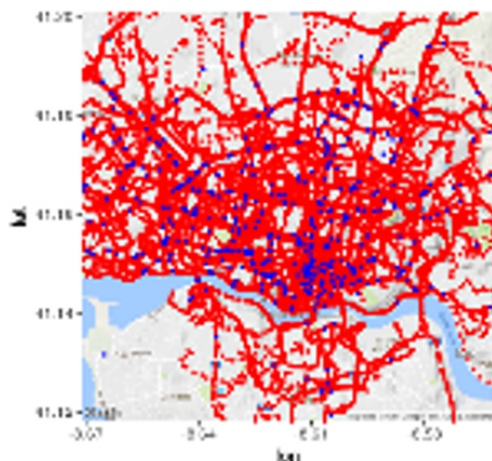


Figure 2: Location Clustering: Red Locations, Blue Cluster Centroid



Figure 3: Comparison of Sampled GPS sequence (Red) vs Location Cluster Sequence (Blue)

#### D. Validation

We test the model using synthetic test data obtained by using partial trajectories from the test data. This allows us to test the predictions on partial trajectories vs actual travel times of full trajectories.

### IV. RESULTS

Table 1 shows RMSE values for different features on synthetic test data with partial trajectories. Similarly Table 2 shows RMSE values for different features on synthetic test data with full trajectories.

As can be seen from Table 1 and Table 2 the overall error is minimized with increase in length. The error values for prefix and suffix are high for smaller values of  $k$ . However the error values for Sampled GPS points are considerably low even for the lower values of  $k$ . As expected the results on full trajectory is much better compared to partial trajectories. Surprisingly the error value for sampled features goes up with the increasing sample size ( $k$ ).

Table 1: RMSE for Test Data with Partial Trajectory

Length	Prefix	Suffix	Sampled
25	1630.364	2494.8560	701.2098
50	1508.055	1026.7833	697.5338
75	2043.408	640.2154	697.4134
100	1380.124	525.6590	696.7501

Table 2: RMSE for Test Data with Full Trajectory

Length	Prefix	Suffix	Sampled
25	1254.7654	1308.8135	21.68519
50	1189.0399	702.1399	44.14646
75	999.3169	766.8669	54.61509
100	801.9511	531.4057	65.33276

Figure 4, Figure 5 and Figure 6 show error distribution across the test trajectories.

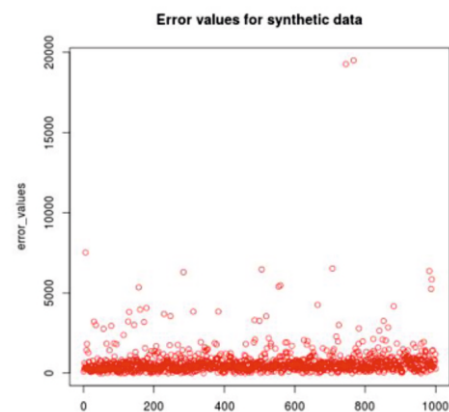


Figure 4: RMSE Distribution for Prefix of Length 100



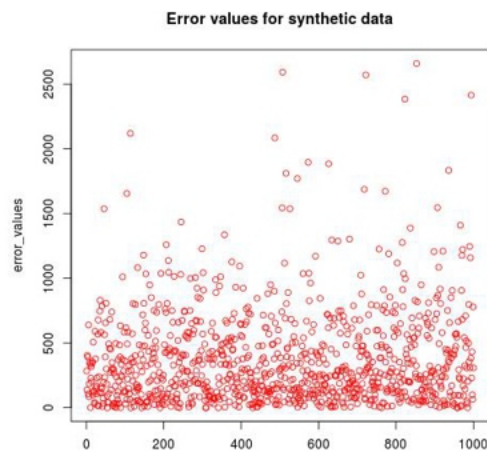


Figure 5: RMSE Distribution for Suffix of Length 100

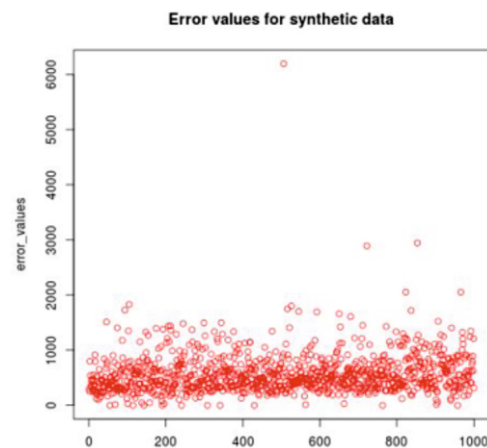


Figure 6: RMSE Distribution for Sample of Length 100

## V. CONCLUSION AND FUTURE WORK

We described our approach to estimate taxi trips using knn regression method. We compared our proposal to use sampled trajectories as a feature with traditional ways of using prefix and/ or suffix of the trajectories. While our approach yields significantly better values with very small number of points, for higher number of features, the results are similar. We also introduced the location clustering based approach to map raw GPS co-ordinates to meaningful locations and then use them for travel time estimation.

The current approach does not use any other features apart from GPS sequence. As a part of future work, we plan to make use of other features such as metadata provided as part of the dataset. Also, we would like to experiment with more complex models such as RNN, HMM which consider sequence of GPS as well instead of just values.

## VI. ACKNOWLEDGEMENT

My first and foremost acknowledgment is to my supervisor and guide Prof. Mr. A. M. Bagul and co-guide Prof. Mr. A. P. Tikar. During the long journey of this study, he supported me in every aspect. he was the one who helped and motivated me to propose research in this field and inspired me with his enthusiasm on research, his experience, and his lively character. I express true sense of gratitude to my guide Prof. Mr. A. M. Bagul and co-guide Prof. Mr. A. P. Tikar for his perfect valuable guidance, all the time support and encouragement that he gave me. I would also like to thanks our head of department Mrs. S. A. Chiwhane, Principal Dr. R. S. Prasad, and management inspiring me and providing all lab and other facilities, which made this project very convenient. I am really thankful to all those who rendered their valuable help for successful completion on project.

## VI. REFERENCES

- [1] A. de Br'ebisson, ' E. Simon, A. Auvolat, P. Vincent, and Y. Bengio, "Artificial neural networks applied to taxi destination prediction," arXiv preprint arXiv:1508.00021, 2015.
- [2] B. Guillouet, L. Jean-Michel, P. Besse, and R. Francis, "Destination prediction by trajectory distribution based model," 2016.
- [3] T. Hoch, "An ensemble learning approach for the kaggle taxi travel time prediction challenge," ECML-PKDD-DCs, 2015.
- [4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional archcture for fast feature embedding," arXiv preprint arXiv:1408.5093, 2014.
- [5] Kaggle, "Pkdd15 taxi trip time prediction," 2015. [Online]. Available:

<https://www.kaggle.com/c/pkdd-15-taxi-trip-time-prediction-ii>

- [6] H. T. Lam, E. Diaz-Aviles, A. Pascale, Y. Gkoufas, and B. Chen, "(blue)taxi destination and trip time prediction from partial trajectories," arXiv preprint arXiv:1509.05257, 2015.
- [7] J. Mendes-Moreira and L. Moreira-Matias, "On learning from taxi-gpstraces," ECML-PKDD-DCs, 2015.